ETH Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Dr. G. Alonso *C. Hagen, B. Kemme*

AR1: Transaktionsverwaltung SS 1997

Praktische Übung 4 Besprechung: 5.6.1997

Der TPC-C Benchmark

In der Übungsstunde wurde der TPC-C Benchmark vorgestellt. Er ist der aktuelle Standard, um die Leistungsfähigkeit von Datenbanksystemen und Rechensystemen unter einer Online-Transaktionslast zu messen. Der Benchmark testet die Leistung aller Hardware-Komponenten wie z.B. Prozessoren, Speicher, Disk I/O, Terminal I/O und der Software-Komponenten wie dem verwendetem Datenbanksystem, Betriebssystem, TP-Monitor etc.

Der Benchmark spezifiziert das Datenbankmodell, die Transaktionen (Arbeitslast) und die Systemumgebung. Es wird eine Lagerhaus-Umgebung beschrieben. Eine Firma besitzt mehrere Lagerhäuser, die Produkte verkaufen. Jedes Lagerhaus verkauft dieselben Produkte und unterhält ein eigenes Lager. Jedes Lagerhaus ist in Distrikte unterteilt, die Kunden verwalten. Diese Information ist in den 5 statischen Tabellen *Warehouse, Item, Stock, District* und *Customer* gespeichert. Die Kunden können Waren bestellen. Dazu wird Information in den Tabellen *Order, New_Order, Order_Line* gespeichert. Das Bezahlen von Ware wird in der Tabelle *History* protokolliert.

Es sind fünf verschiedene Transaktionstypen definiert: (1) Ein neuer Auftrag (*New Order*) geht ein, (2) ein Auftrag wird bezahlt (*Payment*), (3) der Status eines Auftrages wird eingesehen (*Order Status*), (4) die Lieferung zu einem Auftrag wird registiert (*Delivery*) und (5) der Lagerbestand eines Lagerhauses wird überprüft (*Stock Level*). Die Häufigkeit jeder Transaktion im gesamten Transaktionsmix ist vorgegeben. Auch gibt es Vorgaben für die maximale Antwortzeiten etc. Sie sind in folgender Tabelle zusammengefasst:

Transaktionstyp	Min. % im Mix	Min. Tippzeit	90% Antwortzeit	% Min. Denkzeit
New Order	n/a	18 sec.	5 sec.	12 sec.
Payment	43.03	3 sec.	5 sec.	12 sec.
Order Status	4.03	2 sec.	5 sec.	10 sec.
Delivery	4.03	2 sec.	80 sec.	5 sec.
Stock Level	4.03	2 sec.	20 sec.	5 sec.

Dabei entspricht die Tippzeit der Zeit, die für das Ausfüllen der Eingabemaske benötigt wird, und die Denkzeit, der Zeit, die zwischen der Ausgabe einer Maske und dem Aufruf einer neuen Transaktion vergeht. Die 90% Antwortzeit fordert, dass mindestens 90% der Transaktionen des Transaktionstyps in dieser Zeit beendet sein müssen.

Auf unserer ORACLE-Datenbank ist die TPC-C Datenbank mit einer Ausgangsmenge von Tupeln gespeichert und eine Implementierung der Transaktionstypen liegt vor (C mit Embedded SQL). Bitte beachten Sie, dass die im Benchmark vorgegebenen Skalierungen in unserer Testdatenbank nicht ganz eingehalten worden sind. So gibt es z.B. nicht 100.000 verschiedene Produkte, sondern nur 5000. Die Initialisierungen der Tabellen *Order, New_Order, Order_Line* und *History* sind ebenfalls kleiner ausgefallen. Das hat mit Speicher- und Leistungsgründen unseres DBS zu tun.

Die Leistung des Systems unter dem TPC-C Benchmark wird in *New Order*-Transaktionen pro Minute berechnet (**tmpC**).

Sie können sich die gesamte Spezifikation des TPC-C von den WWW-Seiten der Vorlesung holen. Die Home-Page des Transaction Processing Councils hat die Adresse `http://www.tpc.org/home.page.html`. Interessante Information findet sich auch auf `http://www.ideasinternational.com/benchmark/bench.html`.

Zugang zur TPC-C Datenbank und den Transaktionen

Alle Tabellen der TPC-C Datenbank gehören dem Benutzer `tpcc`. Ihre Definition können Sie mit Hilfe des Befehls `describe tablename` unter `sqlplus` erfragen.

Im Verzeichnis ~ar0/p4 finden Sie das Programmpaket zur Generierung und Aktivierung von Transaktionen nach den Vorgaben der TPC-C Spezifikation. Die Datei `TPCtxn.pc` enthält die Implementierung der fünf Transaktionstypen. Die Datei `TPCmain.pc` übernimmt die allgemeine Koordination, sammelt Information über das Verhalten des Systems und emuliert die Terminals, die die Transaktionen starten. Für jedes Terminal wird dabei ein Kindprozess kreiert, der sich mit Oracle verbindet und dann über die gesamte Testzeit Transaktionen startet und ausführt. Die Datei `Record.c` verwaltet CSIM-Statistik-Strukturen.

Das ausführbare Programm erhalten Sie, indem Sie `make` aufrufen. Sie starten das Programm mit `./tpc account password running_time number_terminals print_out analysis debug`. `running_time` gibt die Testzeit an. Die Anzahl der parallel Transaktionen ausführenden Terminals bestimmen Sie mit `number_terminals`. Ist `print_out` grösser Null, so werden die Ergebnisse der Transaktionen in Dateien (pro Terminal eine Datei) geschrieben. Ist das `analysis`-Flag gesetzt, wird Statistik über die Ausführung geführt, wie Sie sie in der ersten Aufgabe kennengelernt haben. Mit der `debug`-Option wird während der Ausführung Information ausgegeben. Am Ende der Laufzeit erhalten Sie pro Terminal die Anzahl der erfolgreich beendeteten Transaktionen je Transaktionsstyp und die jeweiligen Antwortzeiten. Ist Ihnen die Ausgabe zu konfus können Sie sie gerne anpassen. Sie können die Ausgabe auch mittels `> filename` umleiten.

Aufgabe 1: Messen des maximalen Durchsatzes MQTh

Variieren Sie die Anzahl der parallelen Terminals derart, dass Sie maximalen Durchsatz an *New Order*-Transaktionen haben, ohne die Antwortzeitbedingungen der Transaktionen zu verletzen. Welcher Transaktionstyp ist der begrenzende Faktor? Wie weit weicht der Durchsatz von dem in der Vorlesung angebenen maximalen Durchsatz pro Lagerhaus ab?

Erstellen Sie zwei Diagramme bezüglich Antwortzeit und Durchsatz der *New Order* Transaktion, wie sie auch im Abschlussbericht eines offiziellen TPC-C Benchmark Tests gefordert sind:

- Antwortzeitverteilung bei MQTh:
  Die X-Achse gebe verschiedene Antwortzeiten (z.B. 1,2,3 sec. etc). Die Y-Achse misst Anzahl von Transaktionen. Ein Wert (x, y) besagt, dass y Transaktionen eine Antwortzeit zwischen $x - 1$ und x Sekunden hatten.
- Antwortzeit in Abhängigkeit vom Durchsatz:
  Die X-Achse gibt verschiedene Durchsatzwerte bis zum maximalen erreichten Durchsatz an (MQTh). Die Y-Achse ist mit der Antwortzeit beschriftet. Ein Wert (x, y) besagt, dass bei einem Durchsatz von x 90% der Transaktionen in y oder weniger Sekunden beendet wurden.

Versuchen Sie nun, den "Flaschenhals" des Systems herauszufinden. Analysieren Sie dazu die Statistiken, die beim Setzen des `analysis`-Flags ausgegeben werden. Eine andere Möglichkeit ist, in einer eigenen Shell, den Servermanager zu starten (`srvmgrm`). Im Menü `File` erlaubt er ihnen unter dem Menüpunkt `Monitor` diverse Systemtätigkeiten während dem laufenden Betrieb zu beobachten. Inwieweit spielen Konflikte eine Rolle? Bei welchen Tabellen treten am wahrscheinlichsten Konflikte auf?

Aufgabe 2: Isolationstest

Führen Sie einige Tests durch, um zu überprüfen, ob das System die geforderten Isolationsbedingungen erfüllt. Schreiben Sie die Programme geeignet um, so dass jeweils nur die gewünschten Transaktionen mit von Ihnen eingegebenen Parametern gestartet werden und an geeigneter Stelle "angehalten" werden.

- **Lost Update**: In der Übung haben Sie gehört, dass es in Oracle zu Lost Update kommt. Wie und wann kann es bei der parallelen Ausführung der Transaktionen *New Order* und *Delivery* bezüglich des Kunden-Tupels zu einem Lost Update kommen? Wie kann dieser Fehler vermieden werden?
- **Read/Write Konflikt**: Folgender Test ist der TPC-C Spezifikation entnommen:
  1. Start a *New Order* transaction T1.
  2. Stop transaction T1 immediately prior to COMMIT.
  3. Start an *Order Status* transaction T2 for the same customer used in T1. Transaction T2 attempts to read the data for the order T1 has created.

4. Verify that transaction T2 waits.
5. Allow transaction T1 to complete. T2 should now complete
6. Verify that the results from T2 match the data entered in T1.

Wird sich Oracle so verhalten? Wenn nicht, sind die Ergebnisse trotzdem korrekt, wenn man annimmt, dass die Serialisierungsordnung von parallel ausgeführten Transaktion keine Rolle spielt?

- **Write/Write Konflikt**: Folgender Test ist der TPC-C Spezifikation entnommen:
  1. Start a *New Order* transaction T1.
  2. Stop transaction T1 immediately prior to COMMIT.
  3. Start another *New Order* transaction T2 for the same customer as T1.
  4. Verify that transaction T2 waits.
  5. Allow transaction T1 to complete. T2 should now complete
  6. Verify that the results from T2 match the data entered in T1.
  7. Verify that the order number returned for T2 is one greater than the order number for T1. Verify that the value of `D_NEXT_O_ID` reflects the results of both T1 and T2, i.e., it has been incremented by two and is one greater than the order number for T2.

  Wie verhält sich hier Oracle?
- **Phantom**: Folgender Test ist der TPC-C Spezifikation entnommen:
  1. Start an *Order Status* transaction T1 for a selected customer.
  2. Stop T1 immediately after reading the *Order_Table* for the selected customer. The most recent order for that customer is found.
  3. Start a *New Order* transaction T2 for the same customer and complete it.
  4. Continue transaction T1 by repeating the read of the *Order_Table* for the selected customer.
  5. Verify that the order found is the same as in step 3.
  6. Complete and commit transaction T1.

Aufgabe 3: Tuning des Systems

Sammeln Sie Vorschläge, wie die Leistungsfähigkeit des Systems gesteigert werden könnte. Ein Teil wird sich dabei auf die physische Realisierung der Tabellen beziehen, und kann deshalb von Ihnen nur theoretisch durchgeführt werden. Doch auch die Implementierung der Transaktionen kann verbessert werden. Die Beschreibung, was jede Transaktion im einzelnen für Operationen durchführt, kann in der Spezifikation nachgesehen werden, falls es aus dem Programm nicht ersichtlich ist.